

SINC LINK IS A PUBLICATION OF THE TORONTO TIMEX-SINCLAIR USERS CLUB AND IS ISSUED 6 TIMES A YEAR. COPIES OF THE NEWSLETTER ARE \$1.50 EACH FOR NON-MEMBERS. CLUB MEMBERS RECEIVE A FREE COPY AS PART OF THE \$20.00 ANNUAL MEMBERSHIP FEE.

NEWSLETTERS ARE EXCHANGED, FREE OF CHARGE, WITH OTHER TIMEX-SINCLAIR USERS GROUPS

ALL MATERIAL IS IN THE PUBLIC DOMAIN AND CAN BE REPRINTED, PLEASE CREDIT THIS PUBLICATION AND THE AUTHOR IF YOU COPY MATERIAL.

SEND CORRESPONDANCE TO:

Attention: SINC-LINK Editor
TORONTO TIMEX-SINCLAIR USERS CLUB
P.O. Box 7274 Stn. A, Toronto M5W 1X9

INDEX OF CONTENTS

LETTER FROM THE EDITOR	Page 2
PLAYING WITH ELECTRICITY-QL	Page 2
DIGITAL IMAGIZING-Part II by Eric Michaud	Page 3
THE T/S 2040 PRINTER by C. Urban	Page 4
CALLING USR ROUTINES by Dave Hoshier	Page 5
COMMUNICATIONS-BBS by Roelof Mulder	Page 6
SAVING A HEADERLESS FILE	Page 7
BBS FOR TIMEX SINCLAIR USERS	Page 7
CHANGING SCREEN ATTRIBUTES	Page 7
SAVING SCREEN\$ ON LARKEN by G. Chambers	Page 8
POWER SUPPLY FOR THE LARKEN by G. Chambers	Page 8
ADVENTURERS TAKE NOTE	Page 8
ZX81 LARKEN DRIVE SELECT by J. Thomas	Page 9
SMTIH-CORONA FASTEXT 80 by R. Mulder	Page 10
MACHINE CYCLES by M. Lemyre	Page 10
BLOCK PRINTING WITH QUILL	Page 11
GAMES TIPS	Page 11
BOB'S NOTEBOOK by R. Mitchell	Page 12
RE-INKING PRINTER RIBBONS	Page 13
GLEANINGS FROM OUT OF TOWN MEMBERS	Page 13
QUESTIONS! QUESTIONS! QUESTIONS	Page 14
MURPHY'S LAWS (cont'd)	Page 14

EXECUTIVE OFFICERS

PRESIDENT: Reginald Cottle (416) 785 3987
SECRETARY: George Chambers (416) 751 7559
TREASURER: Charles Urban
ACTIVITIES DIRECTOR: Doug Carter
NEWSLETTER EDITOR: Noreen Cottle
HOUSE CHAIRMAN: HELP WANTED!!
LIAISON OFFICER (Out of town members): George Chambers
TAPE LIBRARIANS: John Mann (2068); Rene Bruneau (ZX81)
PAPER LIBRARIANS: Jim McCormack; Mike Lemyre

LETTER FROM THE EDITOR

Greetings. This is the first time I have edited this Newsletter on my own. It is a bit scary but I have been assured (by those who know, or at least they tell me) that it will be FUN. I am going to take them at their word. It is going to be interesting being the editor of a Computing Newsletter since I really do not know that much about computers (they aren't human, are they? Even though they seem to be reproducing at a phenomenal rate in the Cottle household. I guess that is what you get when you leave two computers unchaperoned at night) First off, I would like to apologize for the LATENESS of this edition of the Newsletter. We are in the process of relocating the Club and with the changeover of editors (from George Chambers to myself) we have become more behind than usual. Secondly: WE NEED ARTICLES!!! Something, anything. Do not think that you cannot write or have nothing interesting to contribute because you all have something that would be of interest to at least one other member of the club. Anything that you have discovered while using your computer that made life more interesting, easy, difficult or frustrating will be of interest to us. I am also appealing to our regular contributors, you know who you are, who keep the Club fully informed about what is happening out there in Computer Land. Please don't forget to get your articles to me. Thanks to those people who did submit something and to those who keep meaning to.....WE NEED YOU!!!! NEXT DEADLINE: SECOND MEETING IN MAY (that is May 20th)

Noreen Cottle, Editor

Playing with Electricity

Or - I thought that they said the QL had an RS232 port.

By Harvey Taylor

Vancouver Sinclair UG

726 West 17th

Vancouver, BC V5Z 1T9

There is a problem with the serial ports on the QL. Sinclair calls them full, bidirectional RS-232 interfaces & one would hope that meant everything it was supposed to, but Sinclair cheapness has struck a low blow. Instead of doing the proper thing and using a full UART, Sinclair uses the 8049 Intelligent Peripheral Controller (IPC) for input and part of a custom Undifferentiated Logic Array (ULA) for output.

The problem arises from the fact that the 8049 has a lot to do and cannot watch the input line all the time. It is thus liable to miss characters. To get around this Sinclair has implemented the DTR pin; when the 8049 is busy it pulls DTR low. Any hardware wishing to talk to the QL has to monitor DTR. Most MODEMs are not set up to do this; consequently most MODEMs show a lot of errors.

A further problem arises from the fact that, while baud rate and parity are under program control, the number of stop bits is fixed at two. Most North American BBS systems use one. These factors make it difficult to set up any of the simpler kinds of MODEMs, such as Steve Ciarcia's TMS99532 based project.

My reason for writing this is to search out any solutions other people have tried in trying to circumvent these restrictions. If you have a MODEM working on a QL with any degree of success, drop me a line.

PAGE 2

SINC-LINK

Here is what I have tried. My first attempt was simply to stick an ultra simple TMS99532 MODEM on the serial ports. This resulted in such a high level of errors in reception and transmission that I was unable to log on to any local BBS. (Sigh)

My second attempt was to design a slightly more complicated MODEM incorporating two UARTs. One UART was set to two stop bits and 300 baud like the QL, the other set to the parameters of whatever BBS I was trying to connect to. Once data was available on one UART, the data available pin would go high, triggering a one shot strobe to the write pin on the other UART as well as clearing the data available flip-flop on the first. This still didn't connect.

My third attempt was to utilize the DTR line as well as the data available pin on the second UART to strobe the data into the first UART which ran at 9600 baud. The effect of this is to use the first UART as a one byte buffer. Surprisingly this shortcut works to a degree. Now, however, the original board is so hacked up that I will have to build another. At the present the device is still too unreliable to implement Xmodem.

It would seem that the next level would be to design a smart buffered MODEM which is capable of monitoring the DTR line on the QL. This will involve an extensive project with a dedicated controller chip like the Z8 or suchlike. I've not yet started work on such a device, but, coincidentally, the two QL MODEMs for sale in England incorporate a form of buffering between the MODEM and the QL.

If you have any luck in this endeavour, let me know!

HT

[Tom Bent says there is an i/f available from England @ \$50 to correct this problem. MF]

Digital Imagizing - Part II

By Eric Michaud

Hopefully by now you have assembled the hardware involved in this project and have obtained the necessary driver software from the club library. The software I have written allows you to do high-contrast scans, grey-scaling (the combining of several images giving one picture with varying shades of grey representing the different colours), cleaning of noise in the image, flips, inversions, and cut and pasting from one image to another. If you have not yet obtained this software, I have included a short machine code program that will allow you to capture some high-contrast scans.

Before actually trying out the software, you can check the operation of the hardware with a short basic program. Type in the following:

```
10 OUT 0,255
20 POKE 23692,255: PRINT IN 0;" ";: GOTO 20
```

RUN the program and hook up your STILL video source to the digitizer. Now try turning the BRIGHTNESS and SYNC knobs to various positions. You should get four different numbers in a random order. Usually they will be from 252-255. If you get this then the hardware is more than likely in working condition.

Now load the driver software. Either load the version from the club library or type in the one in this article. If you have the good version, the rest is easy. Just choose option #1 and turn the SYNC knob until the message reads 'IN SYNC'. Now choose a capture option and start digitizing!

If, however, you are using the version printed here, you will have to sync your video source 'by hand'. Connect your video source and RUN the program. Turn the SYNC and BRIGHTNESS knobs until you see a picture starting to form from left to right. If the picture is totally black or totally white, adjust the BRIGHTNESS knob accordingly. If the picture freezes or goes too quickly (shorter than 5 seconds), adjust the SYNC knob. For now this will be more or less trial and error. After this step is completed and you can see the picture from your video source on your 2068 display, you have a working video digitizer!

How does it work? Well, first you must understand TV's a bit. A picture on a TV starts forming from the top left corner and works its way across to the right corner and then returns to the position immediately underneath the top left and so on until it reaches the bottom of the screen at which time the process restarts. A 2068 is not fast enough to collect all this information at once so it only grabs one vertical line of data from each frame. Since there are 60 frames a second and we need 256 lines, this process requires about 5 seconds. There is of course much more to it than this but that is basically it.

The SYNC knob is necessary because of the varying signal strengths from different video sources. Once you have adjusted it for your particular video source, it should not have to be touched again unless you switch to a different source.

The STRECH knob allows you to compress or expand the image. This is done by changing the sampling rate to a slower or faster rate and allows greater detail to be captured.

The BRIGHTNESS knob determines the amount of black or white in the picture. The brightness can also be controlled by software by setting bits 0,1, and 2 of port 00 to a value from 0-7 using binary. Resetting bit 4 restarts the timer in the digitizer. This must be done before each capture. Reading bit 0 provides synchronization information and reading bit 1 gives the actual data for the picture, a one meaning black and a zero meaning white.

I am presently working on some software that will provide 512 by 192 resolution as opposed to the current 256 by 192. This should also be printable on a real printer in the same resolution as well as viewed on the screen.

If you are having difficulties with your digitizer or have questions to ask, feel free to call me at (519) 542-4464. Happy digitizing!

5 REM DIGITIZER LISTING by Eric Michaud

```
10 CLEAR 38911: LET t=0
20 FOR x=38912 TO 39423
30 READ n: POKE x,n: LET t=t+n

40 NEXT x
50 IF t=49564 THEN GO TO 100
60 PRINT "Error": STOP
100 RANDOMIZE USR 39296: BEEP .
01,30: GO TO 100
3000 DATA 0,64,0,65,0,66,0,67,0,
68,0,69,0,70,0,71,32,64,32,65,3
2,66,32,67,32,68,32,69,32,70,32
,71,64,64,64,65,64,66,64,67,64,
68,64,69,64,70,64,71,96,64,96,6
5,96,66,96,67,96,68,96,69,96,70
,96,71,128,64,128,65,128,66,128
,67,128,68,128,69,128,70,128,71
,160,64,160,65,160,66,160,67,16
0,68,160,69,160,70,160,71
3010 DATA 192,64,192,65,192,66,1
92,67,192,68,192,69,192,70,192,
71,224,64,224,65,224,66,224,67,
224,68,224,69,224,70,224,71,0,7
2,0,73,0,74,0,75,0,76,0,77,0,78
,0,79,32,72,32,73,32,74,32,75,3
2,76,32,77,32,78,32,79,64,72,64
,73,64,74,64,75,64,76,64,77,64,
78,64,79,96,72,96,73,96,74,96,7
5,96,76,96,77,96,78,96,79,128,7
2,128,73,128,74,128,75,128,76,1
28,77,128,78,128,79
```

3020 DATA 160,72,160,73,160,74,1
60,75,160,76,160,77,160,78,160,
79,192,72,192,73,192,74,192,75,
192,76,192,77,192,78,192,79,224
,72,224,73,224,74,224,75,224,76
,224,77,224,78,224,79,0,80,0,81
,0,82,0,83,0,84,0,85,0,86,0,87,
32,80
3030 DATA 32,81,32,82,32,83,32,8
4,32,85,32,86,32,87,64,80,64,81
,64,82,64,83,64,84,64,85,64,86,
64,87,96,80,96,81,96,82,96,83,9
6,84,96,85,96,86,96,87,128,80,1
28,81,128,82,128,83,128,84,128,
85,128,86,128,87,160,80,160,81,
160,82,160,83,160,84,160,85,160
,86,160,87,192,80,192,81,192,82
,192,83,192,84,192,85,192,86,19
2,87
3040 DATA 224,80,224,81,224,82,2
24,83,224,84,224,85,224,86,224,
87,243,62,8,0,0,0,211,0,33,0,0,
34,165,153,22,38,205,229,153,21
,32,250,30,128,22,192,205,229,1
53,33,0,152,78,35,70,229,33,32,
0,9,219,0,203,71,32,250,203,79,
40,8,123,238,255,71,126,160,24,
2,126,179,119,225,35,21,32,222,
203,27,48,210,58,165,153,60,50,
165,153,254,32,32,197,62,0,211,
0,6,4,14,255,62,255,61,32,253,1
3,32,248,16,244,251,201,219,0,2
03,71,32,250,62,4,61,32,253,219
,0,203,71,32,239,6,4,62,150,61,
32,253,16,249,201

THE T/S 2040 PRINTER

by
Charlie Urban

Many of you may have been wondering what's inside the funny shaped black box at the end of the printer cable, the one that plugs into your computer, and also inside the large black box that houses the T/S 2040 printer manufactured by Alphacom. Well, here is some enlightening information that I have extracted from Cameron Hayne's article (Sinc-Link July-August, 1985) and from an article in "LIST" the Long Island Sinclair Timex Newsletter of May, 1986. Believe me, that tiny type font in the "LIST" article required a large magnifying glass to decipher the miniature printing (I don't think I'm that old either!).

INTERFACE UNIT

The printer interface unit is the funny shaped black box mentioned above. This is the unit that plugs into your computer and is also connected to the printer box via a short cable.

Most of the control signals and data from the computer's rear edge connector are not used by the 2040 printer but pass straight through this unit to a male connector strip to be available for use by other peripherals such as memory expansion boards, disk drives etc. Thus, the interface unit is basically a signal and data feed through unit with the female side connected to the computer and the male side available for additional interfacing. Control and data signals that are used by the 2040 printer are tapped off the feedthrough circuits.

Inside the interface unit is a circuit board with a capacitor, two ferrite tubes through which the printer cable passes, and an integrated circuit (IC) chip. The purpose of the capacitor and ferrite tubes is to reduce the effects of electrical noise feeding to/from the printer and computer. The purpose of the IC chip, a 74LS10N triple 3-input NAND gate, is to deliver a "pay attention" signal to the printer to tell it when the computer is talking to it.

The schematic diagram of the interface unit is shown in Figure 1. For clarity, only those circuits pertinent to the printer are shown; the straight feedthrough circuits from the female to male connector have been omitted.

TECHNIQUES FOR CALLING USR ROUTINES

by Dave Hoshier

If you've examined some of your programs for any of the Sinclair computers containing machine code, you've probably seen machine code routines called in a variety of ways. The most common calls are:

```
RANDOMIZE USR n    or
LET A = USR n
```

where "n" represents the address where the machine code is located. How do you know which way to call a USR routine? The purpose of this article is to explain why USR routines are called in the ways they are, and to suggest some alternate ways of calling USR routines.

THE BASICS

As you probably know, the heart of the Sinclair computers (QL's excepted) is the Z80 microprocessor. When using BASIC in our programs, we are communicating with the Z80 in an indirect way. Our commands are "interpreted" by the ROM into instructions that are meaningful to the Z80 microprocessor. The extra step of interpreting the BASIC program takes time and means that BASIC programs run slower. Sinclair computers allow us to execute programs that are directly executable by the Z80 microprocessor by use of the USR function. The USR function removes us from the safety of BASIC, since you can easily cause the computer to crash. Don't worry, you can't hurt the computer, but you can easily lose a program painstakingly entered into the computer. You are working without a net when you run machine code programs. The results are breathtaking when the program works, but until your programs are fully debugged, you are only a step away from disaster. For this reason, SAVE your machine code programs before running them.

When invoking the USR function, we are telling the computer to execute a machine code program at a specific address. Inside the Z80 microprocessor, there are three "register pairs" that can hold any integer between 0 and 65535. These register pairs are called the HL, DE, and BC registers (There are other registers, but they are more difficult to use). When we write machine code programs, we manipulate the contents of these registers, and the contents of RAM to some useful purpose.

WHAT HAPPENS ON RETURNING TO BASIC FROM A USR ROUTINE?

In most cases, after finishing a USR routine, we return to BASIC. On returning to BASIC, THE CONTENTS OF THE BC REGISTER PAIR ARE INTERPRETED AS A NUMBER. This fact allows us to pass information from our machine language routines back to BASIC.

Let's write a program that does nothing but load the BC register pair with the number 50 decimal, and return to BASIC. (This will work on any Sinclair computer).

ENTER:

```
POKE 30000,1
POKE 30001,50
POKE 30002,0
POKE 30003,201
```

To the Z80 microprocessor, this program means, load the BC register with 50, then return. In Z80 mnemonics this is abbreviated as:

```
LD BC, 32 (that's 50 in hexadecimal)
RET
```

Now let's prove that the contents of the BC register are interpreted as a number on return to BASIC.

ENTER:

```
PRINT USR 30000
```

The computer will print 50.

Try entering:

```
LET A = USR 30000
PRINT A
```

The computer prints 50, but this time the contents of the BC register have been assigned to the variable A.

A common USR call is:

```
RANDOMIZE USR n
```

The effect of this call is to put the contents of the BC register in the system variable SEED. SEED is used to provide a number for random numbers. To see the results of RANDOMIZE USR 30000 enter:

(TS1000, TS1500, or ZX81 version)

```
PRINT PEEK 16384 + PEEK 16435 * 256
```

(TS2068, or SPECTRUM version)

```
PRINT PEEK 23670 + PEEK 23671 * 256
```

Again, the computer prints 50, the number that has been stored in SEED. (You can find the system variables by looking in the appendix of your manual)

You should be careful when using the RANDOMIZE USR n command. If you are using random numbers somewhere in your program as in a game, the RANDOMIZE USR n command may keep putting the same number in SEED time and time again. This will mean that you will keep getting the same random numbers again and again. After using RANDOMIZE USR n, you should have another line that says RANDOMIZE or RANDOMIZE 0. This will cause the number of frames shown since you turned your computer on to be deposited in SEED. This number should be fairly random.

In the preceding examples we have seen three common ways of calling routines. Now let's examine a couple of more advanced methods.

ADVANCED USR CALL TECHNIQUES

Since we have seen that the BC register pair is interpreted as a number on return to BASIC, it is possible to use USR calls in any way that a numeric argument is valid. For example, you could have a line that said:

IF USR n = 10 THEN GOTO 1000
and it would be perfectly legal. You could execute
multiple USR routines with a statement such as:

LET A = (USR n or USR m or USR p)
and all three routines would be executed.

I enjoy storing DATA in strings. String storage of
data has the advantage of being compact, and the
SINCLAIR computers handle strings especially well. My
favourite USR call is:

LET A\$ = CHR\$ USR n

It has the advantage of setting the system variable
DEST to contain the address of A\$(1). This means that
you can store data in strings and easily find the
location of that string for your machine code routine.
(DEST will also point to numeric variables, but they
are quite a bit more difficult to use) The one thing
that you have to remember when using DEST is that you
must restore A\$(1) to some legal character code
between 0 and 255. This is easily done by ending your
routine with:

```
LD HL, (DEST)
LD C, (HL)
LD B, 0
RET
```

This loads the BC register with the character that
is pointed to by DEST. It is much like saying:

LET A\$(1) = A\$(1)

One final tip on calling USR routines. If you need
to set a value to a USR routine from BASIC you can
always POKE it to some unused location in memory, but
here is a bit more elegant way. Pass the number to
memory with the RANDOMIZE function. Suppose you want
to set the number 3 to the routine. Just enter
RANDOMIZE 3 and the number 3 will be put in SEED.
Then call your USR routine and have it read SEED. It's
simple, and will even put two-byte numbers in SEED.
The one number it won't work for is zero.

Once you have learned that the BC register is the
key to returning to BASIC, you will find that your
machine code routines will be more imaginative.

From the Sept 1985 issue of RAMTOP, Cleveland Ohio.
Retyped by G. Chambers

PRINT FREE for the Spectrum

How many times have you wished there was an easy
way to get the equivalent to the 2068 PRINT FRRE
routine when in the Spectrum mode. But there is a
relatively easy procedure. Just enter the following
as a direct command:

PRINT 65535 - USR 7960

This will give the bytes of free Spectrum memory.

Communications

Well, I finally got my hands on a more-or-less
professional BBS communications package and no, it's
not Tinyboard by Flashware. It's a system called
Casboard 2068 (the Flexi-Board System) by Kurt A.
Casby. It is comprised of a block of code that handles
the Xmodem protocol for uploads and downloads and a
block of basic for the normal menus and the like. It
comes on a tape with versions for tape(with no
upload/download), microdrive and disk. To all intents
and purposes, it appears to the caller as a
professional system with multiple message bases,
upload and downloads, chat with the sysop(if he's
there that is) and an extensive set up routine
possibility for the sysop. It even requires a common
password arrangement so that only those to whom you
have provided the same, can use the BBS.

I had a number of occasions to test the BBS system
with a number of my friends and all except for the
auto save to disk of an uploaded program the BBS
operated as advertised; ie perfectly. The problem of
the upload to the disk appears to be more of a problem
with the caller's software (Specterm-64) rather than
the BBS program, but that has as yet to be proven.

There is a standard setup routine that the sysop must
follow in order for the program to function correctly.
If this process is not followed to a 't', then the
system will stop with an error that is usually good
enough to stop the program in it's tracks. The sysop
must identify the password, upload and download
permissability, message base usage and a few other
misc items. After doing this for the first time and
subsequently saving to disk, future setups can be
quite quick. The sysop then sits back and watches TV,
or to some poor slob routing around through the BBS.
You can always force a chat routine with the user if
the routing becomes crazy.

I had thought about setting up the system on a
special time and date routine for my local club and
had thought that the best hours would be between 8 and
10 p.m. on a Saturday and Sunday evening,
particularly because of the phone ringing, or so I
thought. What I subsequently found out was that the
BBS program answers the phone so fast that the
telephone doesn't have a chance to ring. This is great
to one point of view, but it was particularly
disconcerting to my mother-in-law to receive the
rather harsh tones of a high frequency modem rather
than the sweet melodic tones of my voice. She called
me 2 hours later with a few words to the wise.

In any case, I am, through all this rambling,
recommending this Casboard BBS system to all who would
like to try their hands at being a sysop. Great fun
and a sense of accomplishment!
Roelof Mulder

HOW TO SAVE A HEADERLESS FILE
using a machine code routine
(TS 2068)

First load the A register with 255. This is to signify that you're saving a block of data and not a header. Then load IX with the start location of the block of code you intend to save. Now load DE with the number of bytes that you want to save. Then CALL 04C2h, and RET brings you back to Basic.

Here's an example, if you're feeling a bit overwhelmed. Say you want to save a screen picture without a header. Then your assembler listing would read something like this:

```
LDA,255
LD DE,6912d
LD IX,16384d
CALL 04C2h
RET
```

Now, to reload the screen picture you've just saved, the code reads as follows:

```
SCF
LD A,255d
LD IX,16384d
LD DE,6912d
CALL 0556h
RET
```

The parameters here are the same as when you're loading, except for the SCF command which sets the carry flag. If we didn't set the carry flag, the program would just verify instead of loading. If you want to do a headerless verify, just swap the SCF command for an AND A instruction.

from Your Spectrum June 1985 p11.
retyped by G. Chambers

TS2068 TIPS

POKE 23730 & 23731 WITH RAMTOP
**RESETS RAMTOP W/O DESTROYING
VARIABLES THAT ARE NOT DEFINED
IN THE PROGRAM.

1 POKE 23672,0: POKE 23673,0
9999 PRINT INT ((PEEK 23672+256*
PEEK 23673+.5)*100/60.1145+.5)/1
00
**GOTO 9999 TO FIND THE ELAPSED
TIME IN SECONDS THAT PROGRAM
HAS BEEN RUNNING.

POKE 23561,0
**STOPS KEY REPEAT. MAY BE USE-
FUL IN GAMES & UTILITIES.

POKE 23561,1
**USE TO DISABLE KEYBOARD AS
COPYRIGHT PROTECTION.

BBS For Timex-Sinclair Users

The SYSOP of 'DOWNLOADERS ANONYMOUS' has kindly provided the Toronto Timex-Sinclair Users' Club with both MESSAGE and FILE sections for uploading/downloading public domain software. Both in and out-of-town users are urged to call to become validated as soon as possible, and start using the services provided to them!

This bulletin board system also features an OUTSIDE FIDO command, BUY&SELL, JOKES, and GENERAL MESSAGE SECTIONS, as well as FILE AREAS for IBM, APPLE, AMIGA, and ATARI computers. Until March 31, 1987, all areas are available to all users. A nominal contribution of \$10. will be required to access areas other than the message areas after this date.

For those who do not yet have a modem, untested boards from Grey and Clifford are still available for \$6. through Tony Mazza.

Remember to get your modems working and call 844-2035 TODAY! There may be telecommunications seminars given in the future. Get ready!

TO CHANGE SCREEN ATTRIBUTES WITHOUT AFFECTING DISPLAY

The following Machine code routine will change the whole screen's attributes without disturbing the display, and is faster than it's BASIC equivalent. The routine can be located anywhere in memory.

62	56	LD A,56	To change colour, change 56 to ATTR value required
33	0	88 LD HL, 22528	Start of Attribute file
17	1	88 LD DE,22529	To be copied into
1	255	2 LD BC,767	Number of bytes to fill
119		LD (HL),A	Fill the byte
237	176	LDIR	Load Increase Return till BC=0.
50	141	92 LD (23693),A	Copies ATTR into ATTR P (Permanent Attributes)
201		RET	Return to BASIC

To change Attribute colour, (currently black ink, white paper) poke the start address+1 with Attribute value.

From the Feb 1986 newsletter of the ZX Mushroom Club.

SAVING SCREEN\$ ON THE LARKEN DRIVE

by George Chambers

Although we often do not bother to save the SCREEN\$ portion of a program to disk, since it's appearance is so transient, there are occasions when saving a SCREEN\$ display is desirable. The following program will facilitate the process.

The program is very straight-forward, however one or two points should be noted. LINE 10 is used to SAVE the SCREEN\$. A successful SCREEN\$ SAVE depends on masking the LDOS cursor which normally appears on the screen during a DOS operation. We do this by ensuring that the DOS cursor positions itself in a plain background area of the SCREEN\$ to be saved, and by making the PAPER and INK the same color as the background.

This will require that you load the SCREEN\$ first to determine where the LDOS cursor should be placed.

The listing below was used on a program where there was suitable length of magenta background on the left-hand side of line 1, therefore "PAPER 3:INK 3:PRINT AT 1,0;" was appropriate. Revise LINE 10, and also LINE 20; according to the need of the particular SCREEN\$ to be saved.

Once this SAVE program has been loaded and revised per above, RUN it, place a disk in your drive and load the SCREEN\$ from tape. When SCREEN\$ finishes loading this program will immediately save it to disk. To verify, press any key, and the SCREEN\$ will reload.

Lines 20 and 30 are used to load the screen display back into the computer, and are included here so that a successful SAVE can be verified. Normally they would be used in another program loading routine. Note the use of the same "PAPER 3: INK 3: PRINT AT 1,0", as was used to SAVE the display initially.

```
10 CLEAR 61000: OUT 84,64: POK
E 23540,0: POKE 23541,64: POKE
23542,0: POKE 23543,27: CLS : L
OAD ""CODE : PAPER 3: INK 3: PR
INT AT 1,0;; PRINT USR 63488: R
EM save"sweevo.C2"
```

```
15 PAUSE 0
```

```
20 PAPER 3: INK 3: CLS : PRINT
AT 1,0: PRINT USR 63488: REM 1
oad"sweevo.C2"
```

```
30 PRINT #1;AT 1,5; FLASH 1; I
NK 7; PAPER 0;"Press any key to
start": PAUSE 0
```

When I first purchased my LARKEN interface I went shopping for a power supply for it. I came across a surplus unit for \$22 and bought it. While setting it up I think that I inadvertently damaged it, for the 12V presently read about 6V. However, presently this condition corrected itself and I proceeded to use it on the system.

However, after some 6 months I started to get CRC error messages with my drive, for no apparent reason. I checked the power supply and found that I had considerably less than the required 12V. My power supply was defective.

Rather than mess around (it was a switching power supply, with about 50 odd components to it) I went out and bought the parts for another power supply, from Radio Shack. (This was Larry Kenny's recommendation from the start, incidentally).

I bought the following parts from Radio Shack:

Item No.	Quant.	Part No.	Name
1	1	276-1771	12V Regulator - 7812
2	2	276-1770	5V Regulator - 7805
3	2	272-1016	100mfd 35V Elec. Cap.
4	1	272-1020	2200mfd 35V Axial Cap.
5	1 pkt	276-1101	1 Amp. 50 Piv Rectif.
6	1	273-1512	Transformer
7	2		HEAT SINKS

This cost a total of \$32. I already had a housing for the assembly, and the power cables, connectors, and fuse holder. I simply wired the items up per Larry's schematic, and now all is well.

ADVENTURERS TAKE NOTE

One of our members has asked that we insert this little item: GFC

ADVENTURERS!! Join the fastest-growing hint trading group in North America. Send SASE with hints, questions, and problems, to Douglas Jeffery, Larch Road. R.R. #1, Telkwa, British Columbia, V0J 2X0

ZX81 LARKEN DRIVE SELECT ROUTINE

by John G. Thomas

The program listing accompanying this article selects either drive 1 or drive 2 for the LARKEN DOS, from any MC storage area in the TS1000 (ZX81) where you can safely place M/C. This can be the 8-16K area (you must have RAM here, either by having the 64K memory or a Hunter board). POKE the M/C provided, beginning with 8192; or above RAMTOP (after establishing a proper RAMTOP by POKEing the necessary numbers in 16388 and 16389 in the TS1000 (don't forget to NEW), POKE the M/C beginning with the first address of the of the selected RAMTOP; or in a REM statement (generate it first, you will need 18 characters in it, and POKE M/C beginning withb 16514); or any other place suitable for M/C.

If the M/C has been entered beginning with 8192, then the call for drive 2 will be RAND USR 8192, and for drive 1 RAND USR 8205.

If the M/C is in a 1 REM statement then the call for drive 2 will be RAND USR 16514, and drive 1 will be RAND USR 16527.

This program can be used with the 2068 if changes are made regarding the M/C storage areas (address of the M/C storage areas determines where to place and where to call the program). Bytes that are POKEd by the M/C to select the drives in the ZX81 are not the same as for the 2068. These bytes are: 8193- 87 (the port number); 8194- 2 (data for the port); these bytes select drive 2. Bytes 8206- 87, 8207- 0 are likewise the port and data for drive 2. These bytes may have to be changed for the 2068. Check your DOS manual, regarding drive select.

The bytes at 8202- 0 and 8203- 56 call LDOS at 14336, and also may have to be changed if the call is different for the 2068.

I use the program to select the drives; one has utility programs that are all placed at 8192 and all are in M/C. (Use the M/C SAVEing technique of LDOS). Now, I only have to remember one address and I can load as many M/C utilities as I want, one after another from 8192 and no conflict with the resident BASIC program. The second drive is for storage of the BASIC program as I progress (frequent SAVES prevents suicides among programmers!).

FOR SALE

1 - Timex 1000 Computer, equipped with a large keyboard.

1 - Timex 2040 Printer.

1 - Basic Course on the TS 1000

A great many software tapes, commercial

and public domain.

Write to Andrea Stub, 262 William St., London, Ont.,
N6B 3C3.

SINC-LINK

ADDRESS	DEC	HEX	COMMENTS
8192	1	01	Call for Drive #2 (8192)
8193	87	57	
8194	2	02	
8195	237	ED	
8196	67	43	
8197	12	0C	Both leave you at the
8198	48	30	entry to DOS.
8199	237	ED	
8200	65	41	
8201	205	CD	
8202	0	00	
8303	56	38	
8204	201	C9	
8205	1	01	Call for Drive #1 (8205)
8206	87	57	
8207	0	00	
8208	237	ED	
8209	67	43	REMEMBER to poke 12304,0;
8210	12	0C	POKE 12305,32; POKE 12306, 26
8211	48	3	and POKE 12307,0 before saving
8212	237	ED	on disk with "Drive.C1"
8213	65	4	whatever name.
8214	205	CD	
8215	0	00	This M/C can be located at
8216	56	38	any convenient spot.
8217	201	C9	

DRIVE #2 SELECT

1,87,2	Sets up what will be poked in address 12300 and 12301 of the DOS.
237,67,12,48	POKEs data above into 12300,12301
237,65	Uses data in 12300,12301 to select port.
205,0,56	Enters DOS
201	Return to BASIC

DRIVE #1 SELECT

1,87,0	Sets up what will be poked.
237,67,12,48	Pokes data into addresses 12300 and 12301
237,65	Uses data in 12300 and 12301 to select port.
205,0,56	Enters LDOS
201	Return to BASIC

Smith Corona FASTEXT 80
Dot matrix Centronics printer

Cost: \$239.00 Cdn.
Available from : Ogilvies

Remarks:

After many years of using the 2040 Timex printer, I finally broke my wife down and went out and bought a real honest to goodness dot matrix printer. I didn't hurt her much, I swear!

The obvious question that I ask myself now is, How did I do for so long without? I was even using my 2040 to send letters to companies and I even got a query concerning the quality of toilet paper that I was using! Imagine that.

To the review..

The reason I purchased this particular printer was primarily due to the price. You have to admit that \$239.00 is a good price. The second reason is that I was witness to a demo of the printer at the local Ogilvies. The printer was connected to an Apple IIC and was showing off its graphic printing facilities. Impressive to say the least! Not only could it pump out graphics, but it can type out in Pica, Elite, ENLARGED, condensed, at 5, 6 or 8.3 16.7 characters per inch, and can do graphics at 60 or 72 dots per

inch. Normal text printing is fast with bi-directional printing as standard and uni-directional for graphics. It also has a repertoire of 8 built-in character sets!

The printer comes as friction feed standard and a form feeder is difficult to get. I found a good substitute. Radio Shack sells a printer DMP 105 that overlooking the exterior casing is identical. The tractor feed for the Radio Shack printer fits the Smith Corona like a glove, problem solved. (cost of tractor feed- \$29.00)

The second thing that will be of concern is the print ribbon. Ogilvies sells a dual cartridge package for \$29.00. If this seems a steep price, your right. Instead go to your trusty Radio Shack and pick up one printer ribbon cartridge for the DMP 105 for \$10.00 . Problems all solved!

As you can see from this article, the printer works rather well, wouldn't you say? By the way, the print for this text is in condensed mode, and a ribbon cartridge should last for 1 million characters. There's not much more to know about the printer, but you will be required to find a suitable interface for it and as well, the correct cable. The Aerco, Tasman and Oliger interfaces work very well. If you get the Oliger interface, you will also get the appropriate software to run it. Software packages such as Macscript and Tasword already function without the printer software.

If you would like to learn more about this printer, feel free to contact me through this newsletter.

R.Mulder

DEC.27 1987

MACHINE CYCLES

by Mike Lemyre

I was reading an article in P.C.magazine that pertained to accelerator boards for I.B.M. computers. The article gave a comparison of the number of machine cycles that a particular instruction requires in order to execute fully. I am going to try to list some comparisons between the 8088 and the Z80A CPU. The time required for 1 clock cycle is equal to 1/clock speed. In the case of the Z80A it = $1/3.53e6$ which = 283 nanoseconds. The I.B.M. runs at 210 nanosecs. That means that the 8088-based machine runs 34% faster. However that is not a true assessment of the machine performance because of the need to compare both the display handling and the number of cycles required to complete an op-code. The 8088 based machine feeds a speed of 14.312 mhz to the display card in comparison to the 2068 which uses a 14.112 mhz speed for its video generation. These frequencies compare favorably.

Now let us look at some commands:

	8088	Z80A.
NOP	3	4
JP,N	15	10
ADD A,N	4	4
LD IX	14	14

There are other factors to consider when comparing the performance of machines. Some are the access times of memory chips and other support components. A point to remember; both the 8088 and the Z80A access data in 8-bit chunks. My experiences with large databases reveal that most time is used in disk I/O Graphics... such as drawing boxes or lines. These show no observable gain in performance. The programs that are memory intensive are usually written in a high-level language and compiled. The compiled versions run much faster. Again you will not be able to notice a difference in performance in everyday applications.

Block printing, saving, and loading with QUILL

One of the major failings that I held against QUILL was the fact that it didn't have a command for block printing, saving, and loading. It still doesn't have commands to do this but it can be done even though a little circuitous. The issue is being able to extract a block of text from a QUILL document and then be able to print, save and/or load it into another document. The procedure goes like this:

Let's say that you have a one page document with three paragraphs and you want to select the middle one for printing. First put the cursor on the line above the middle paragraph and use the Page command (see PAGE(II) on page 23 of the QUILL section in the QL User's Guide) on the second (Commands II) menu to insert a page break (looks like ----- across the screen). Next move the cursor to the line after the middle paragraph and use the Page command to insert another Page break. This results in isolating the desired block (in this case paragraph two) into its own page (in this case page 2). Don't forget to hit ENTER or ESC to end the command. With the print command, you can instruct the printer to begin and end at page 2 (instead of the "whole document" option QUILL prompts you with).

To Save the Block, use the Print command and instead of taking the last option "to printer" that QUILL prompts you with, type temp (or some other valid QUILL file name). This will divert the output to your cartridge (assuming you have a formatted tape with sufficient blank space) in mdv2. The file will be called temp_lis rather than temp_doc when you do a directory listing as is usual for QUILL files. To load this new file you have use the Import option of the Files command (on the second, ie Commands II, menu. See FILES II, page 27, QUILL section, User's Guide). This is because the Load command only recognizes valid QUILL files and they end in _doc. By the way, changing the file name from temp_lis to temp_doc doesn't work. One note, when Importing, you will be prompted with "by line". Just hit ENTER and it will work.

When you are ready to Delete the temp_lis file, you can use the Delete option of the Files command (on the second, CommandsII, menu) but you have to specify temp_lis rather than the usual temp because it is not a standard QUILL file and it will tell you so.

This "corrects" the only real deficiency I have found in QUILL and now it is an all round wordprocessor and, for CDN\$15.00, very hard to beat on a performance per dollar basis (there are probably some public domain examples out there). I hope this bit proves useful to you.

ELITE

You can cheat in the game ELITE by pressing Y to load a new commander at the beginning. When the SAVE/LOAD menu appears, press 2 and enter any name. When you press 3 you will find that you have become an Elite. You can then equip your ship properly. After that you can sell all your wares and press 2, whereupon you will be asked if you want to save the people. If you answer Yes they will automatically board your ship. After you have done that you must set your hyperdrive system and you can take off. As soon as you have taken off you must press H or else you will be destroyed by the supernova. When you land at the next planet the people you are carrying will give you 100 g. of gems and you can continue with the game.

From the Feb 1986 newsletter of the ZX Mushroom Club.

TOMAHAWK

The TOMAHAWK simulation is based on the Hughes AH 64A Apache US Army Advanced Attack Helicopter. This first entered service in 1984, and is expected to have 700 units operational by 1990

The realism of this sequel to the excellent FIGHTER PILOT is quite impressive - together with its comparative simplicity of operation.

The graphics are very good indeed, and are claimed to contain over 7000 ground features including mountain ranges, buildings, pylons, trees and other landscape features, not to mention targets, tanks, guns, and helicopters.

The instrumentation is extremely well presented and accurate, with a choice of cannon, rockets, and guided missiles, via the weapons option. Numerous flight options are available to the player, including weather conditions, cloud base, skill level, and day or night flying with an infra-red option for the latter.

Numerous beacons are provided, and the most useful on-board instrument is the Doppler Navigation Radar, which guides you to beacons, bases, landing pads, or targets with absolute accuracy.

TOMAHAWK has merits too numerous to mention and it is an absolute must for any Spectrum owner remotely interested in aviation and simulators. TOMAHAWK should find the same degree of appeal to general gamers as did FIGHTER PILOT. HIGHLY RECOMMENDED!!

From the Feb 1986 newsletter of the ZX Mushroom club.

Ed. NOTE: It will work on the 2068 equipped with the Spectrum ROM.

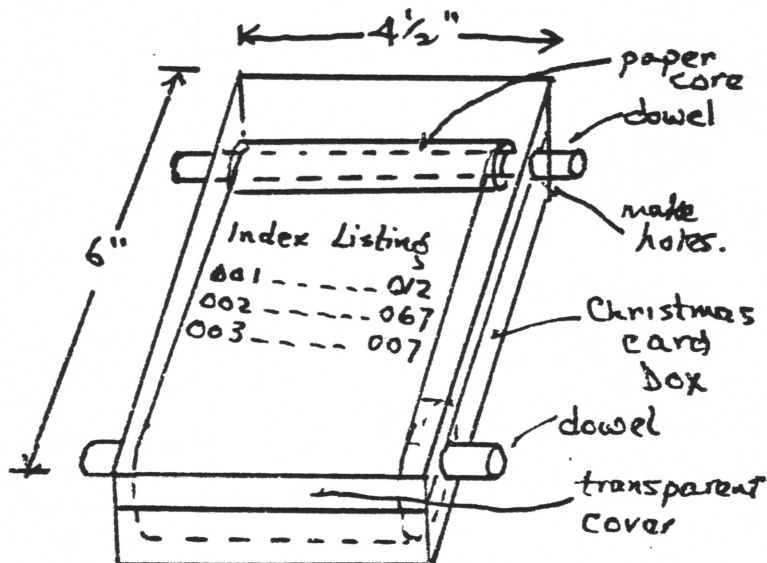
BOB'S NOTEBOOK

This time, I'll talk about a different topic: how to organize your program information.

When you accumulate a large list of programs, either on tape or on disk, you should have a quick way to access the information. Using disk (in my case, Larten) I collect all the titles and disk numbers using the INDEX program which George Chambers & I have been working on. This will accommodate 1000 titles. Now this means about 500 if you make dups of each program.

Once the info is stored in the Index program, you can then get at the list quite quickly by loading it and displaying in on the screen. However, there are times when this is inconvenient (eg, you have a program already loaded in your computer). I have found it better to make a paper copy of the program index using the 2040 printer. Then I roll this onto my handy-dandy home-made holder which I'll try to describe.

I took a Christmas card box (the kind with a transparent plastic cover) and made two holes in the two sides just the size to take some doweling (3/8"). I use two cores from 2040 paper rolls and fit those onto the dowels which are cut to protrude about 1/2" outside the card box. I found that I had to wedge some paper between the dowel and the core to stop the core from turning. With these cores in place, I taped the end of each print-out to the two cores and I was in business. Here is a sketch of my version.



If you are still using tape to save your programs, you can use a program like the one I called PROGRAM57H (or some such name) which is available in the club tape 2068 library. Incidentally I find that if I am using Timex paper (as opposed to the better Radio Shack TP10 paper), I use a darker font style such as the one I am using for this article.

I update my paper list every month or so and in the interim I make a pencil list of any additions or changes just to keep track until I make a new print-out.

This procedure may sound old-fashioned and inconsistent with using a computer, but I have found it very useful and very much the time-saver.

Next item.

Disk-drive users will know that program names usually have an extension of two or more characters. Larten allows you to pick your own second character to suit yourself. I established my own code to keep my program names organized better. This also helps in making sub-lists of your main list using the "find" routine in the INDEX program mentioned above. Here is my code which you would likely tailor to suit yourself.

c compiled program (after C) +
s spectrum
u under development
m master
t text (after A, eg. t\$list.At)
x experimental (like "u")
d alternative to "c" above
p basic ready for compiling +
b boot program (after B)
note + using Timachine

Finally, I always keep track on paper of the date on which I update data in programs like Index, Budget, Ucalc spreadsheet files, graphs on hydro/gas usage and car mileage, etc.

Until next time! Bob Mitchell

JAN. 16 1987

Que....Do you know where I can find a device to re-ink printer ribbons?Query from a reader.

Ans....We know only one company in Canada supplying re-inking equipment. Abba Computer will sell you kits to ink cartridges or spools for under \$130. For the cartridges you have to buy a universal base for \$111 and then any one of dozens of \$16 adapters they have for most ribbons on the market.

Ink will cost you \$8 a bottle--good for anywhere from 10 to 60 applications, depending on the ribbon. ABBA is in Markham at 21 Amber St., Unit 4. Phone 477 4033.

We are told that it takes a bit of practice to get the technique down just right, so you might consider having someone else do it for you. Again, we've come across only one Canadian company offering the service. Robert Jones Enterprises will recycle most ribbons for anywhere from \$5 to \$15. It is at 755 Don Mills Rd., Suite 1702. Phone 444-8695

From TORONTO COMPUTES magazine

GLEANINGS

From correspondence with our out-of-town members

-from Luca Martini, Montreal.

"....I recently bought a Sinclair QL kit. I read in a club newsletter that someone is having trouble with the monitor. I find the picture very good and even better than the IBM high resolution screen. The resolution on a QL is 512 # 256 which is quite considerable, so one should use a high resolution monitor with horizontal bandwidth of from 15 KHz to 30 KHz, and enough vertical lines (300). If the monitor used is not fast enough you get a vibration effect on the characters, as well as other effects. I had a problem on the 2068 which was caused by exactly the opposite situation, namely the TS2068 is too slow for my monitor."



MEYLENSTEYN IN OMNI

Selections from members' correspondence with GFC...

From Larry Crawford, London, Ont.

"...The Oliger DOS is now a winner and still extremely easy to use. I like it a lot. Have just ordered Smart Text from Bill Jones of Gulf Micro Electronics in Panama City, Florida. It is a complete Data Base Management System on Oliger disk. It has all the necessary word processing functions as well as complete printer control for "desk-top publishing". The exciting feature though, is the ability to fetch data from disk. Am looking forward to getting my hands on it. Will let you know how it shapes up."

"...Another interesting project in the wind is coming from Bob Orrfelt in California. He has expanded the EXROM to 16K and has stuffed it with all manner of routines. He has offered to send me one for evaluation..."

"....Oh yes, memory banks. I have 64K in two I/O ports on a 4 1/2 inch square board using eight 6264 (8K by 8) SRAMS. Orrfelt tells me that 32K by 8 SRAMS are available in the US. I can't even find them listed in the Canadian suppliers' new catalogues. Would you nose around for me please? They would probably be 62256's. With eight of them installed, I would have 256K on that one little board. Please don't ask me what I would want all that memory for. I don't have a sane answer. Probably just to hold my own with the guys at work who own Tandy 1000's!!..."

From John Thomas, San Antonio

Topic....ZX81 and the LARKEN Disk System.

"About drives and the LDOS: LDOS needs pin 34 on the disk drive cable to go low when the disk is at speed and ready to process data, but many drives do not even have a connection to this pin at all. Of the drives that do not use pin 34 as a ready line some use pin 6 (drive 4 select pin) as a ready line output. So a fix in this case would be to jumper from pin 6 to 34 on the disk cable (on the DOS pc board is the easiest place to do it). Larry K. says another easy fix is to ground pin 34 all the time, as the DOS has a time delay built in (do this at the fingers of the pc board as above, or ground the upper end of the 150 to 470 ohm resistor standing up at the outside edge of the component side (about half-way), this grounds pin 34. Another problem is to have the drive not set up for MS (MM is not correct); DS instead of MX; RY, RD, and DR jumpers selected if present; and the proper drive selected (DS1 to 4). Also be sure to use the resistor pack only on the last drive."

Questions! Questions! Questions!

Is there anyone out there who knows anything about the 2nd, 3rd, and 4th display modes available on the 2068? Now, some of you probably didn't know there were 4 display modes available on the 2068. Display Mode (DM) 1 is the regular one we're all used to using. (256x192 pixels). DM II has twice the resolution of DM I (512x192 pixels), but has few definable attributes, i.e. the whole screen has to be the same colour. DM III is exactly the same as DM I. While one is flashed on the screen, the other is being updated (redraw), so that we're allowed somewhat crude animation. DM IV has the ability to define attributes on the pixel level, so you could have each pixel exist in a different attribute state. Thus allowing some pretty "hi-res" graphics capabilities.. The only problem is that no one seems to be taking advantage of these wonderful capabilities; there seems to be little software developed which uses these display modes (except maybe MSCRIPT), and certainly no utility programs which aid in the use of these Display Modes. So if anybody has any information on these 4 Display Modes I would be glad to hear about it. Orin Zelanak, R.R.#1, West Lorne, Ontario, Can.NOL 2P0



"It says, 'Leave me alone, I have a headache!'"

MURPHY'S LAWS OF COMPUTING

The number one cause of computer problems is computer solutions.

If an hour has been spent perfecting a sentence, a committee will decide to delete the paragraph.

A committee is a group of persons doing the work of one person.

When you're asked if you need help to understand a computer program, say "no", then negotiate.

The further off new technological developments are, the better they look.

Postmaster, if Undelivered Return to :

Toronto Timex - Sinclair Users Club
P. O. Box 7274 Sin. A
Toronto, Ont., M5W 1X9
Canada

